In the Claims:

Please amend the claims as follows:

1-20. (canceled)

21. (new) A method of mapping a plurality of virtual registers to a plurality of physical registers, the method comprising:

providing a plurality of virtual registers, wherein each of the virtual registers comprises physical register address bits; and providing a status indicator for indicating a status of each of the virtual registers;

designating a subset of the virtual registers as virtual local registers;

executing a save command, the executing of the save command comprising saving a mapping of all of the virtual local registers onto a stack; saving a status as indicated by the status indicator for each of the virtual local registers onto the stack, and setting the status of all of the virtual local registers to "clean";

executing a restore command, the executing the restore command comprising popping the mapping of all of the virtual local registers from the stack to the virtual local registers; and popping the status of all of the virtual local registers from the stack;

binding an argument, the argument binding comprising binding a first virtual register of the virtual registers to a second virtual register of the virtual registers; and binding the status of the first virtual register to the second virtual register;

wherein the argument binding further comprises saving a mapping of the second virtual register onto the stack, saving the status of the second virtual register onto the stack, placing a physical address stored in the first virtual register in the second virtual register, and setting the status of the second virtual register to the status of the first virtual register; and

wherein the argument binding occurs during a call instruction, wherein the call instruction has at least one argument, wherein the first virtual register is used for the at least one argument.

22. (new) A method comprising:

decoding an instruction;

if the instruction is a call instruction, then binding an argument of the call instruction;

wherein the argument binding comprises

copying a first virtual register of a plurality of virtual registers to a second virtual register of the virtual registers, each of the virtual registers comprising physical register address bits, and

copying a first status indicator of a plurality of status indicators to a second status indicator of the status indicators, wherein each of the status indicators corresponds to a respective one of the virtual registers; and

wherein the first virtual register is used for the argument and the second virtual register is used as a formal parameter.

23. (new) The method, as recited in claim 22, further comprising:

mapping a virtual register of the plurality of virtual registers from an old physical register to a new physical register, when the virtual register is a destination virtual register of an instruction being decoded; and

placing an address of the old physical register in an instruction retirement list related to the instruction being decoded if and only if the status indicator corresponding to the virtual register is not clean.

24. (new) The method, as recited in claim 23, further comprising:

saving the physical register address bits held in the second virtual register and the second status indicator to a stack and then setting to clean the second status indicator.

25. (new) The method, as recited in claim 23, further comprising:

setting the second status indicator to not clean when the second virtual register is mapped to a new physical register.

26. (new) The method, as recited in claim 22, wherein:

a subset of the plurality of virtual registers are virtual local registers.

27. (new) The method, as recited in claim 26, further comprising:

executing a save command, the executing of the save command comprising saving the mapping of all of the virtual local registers onto a stack; and saving the status indicators corresponding to all of the virtual local registers onto the stack.

28. (new) The method, as recited in claim 27, wherein:

the executing of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators onto the stack.

29. (new) The method, as recited in claim 28, further comprising:

executing a restore command, the executing the restore command comprising popping the mapping of all of the virtual local registers from the stack to the virtual local registers; and popping the status indicators corresponding to all of the virtual local registers from the stack.

30. (new) The method, as recited in claim 29, further comprising:

selectively executing the restore command if the instruction is a return instruction.

31. (new) The method, as recited in claim 22, wherein:

following the argument binding, if the first virtual register is a destination register, the first virtual register is assigned a first physical register address which is different than a second physical register address stored in the second virtual register.

32. (new) The method, as recited in claim 31, wherein: before the assignment of the first physical register address to the first virtual register, a corresponding first physical register status is “free”.

33. (new) The method, as recited in claim 32, wherein: after the assignment of the first physical register address to the first virtual register, the corresponding first physical register status is “waiting”.

34. (new) A processing device including:

an instruction decoder adapted to decode an instruction;

a plurality of physical registers;

a plurality of virtual registers, each of the virtual registers comprising physical register address bits;

a plurality of status indicators, each of the status indicators corresponding to a respective one of the virtual registers;

wherein if the instruction decoder decodes a call instruction, then binding an argument of the call instruction, the binding comprising

copying a first one of the virtual registers into a second one of the virtual registers, and

copying a first one of the status indicators into a second one of the status indicators, the first status indicator corresponding to the first virtual register and the second status indicator corresponding to the second virtual register; and

wherein the first virtual register is used for the argument and the second virtual register is used for a formal parameter.

35. (new) The processing device, as recited in claim 34, wherein:

if the instruction decoder decodes an instruction having a destination virtual register selected from the virtual registers, then

mapping the destination virtual register from an old physical register of the physical registers to a new physical register of the physical registers, and

placing an address of the old physical register in an instruction retirement list related to the instruction if and only if the status indicator corresponding to the destination virtual register is not clean.

36. (new) The processing device, as recited in claim 35, further including:

a stack; and

wherein if the instruction decoder decodes a call instruction, then saving the physical register address bits held in the second virtual register and the second status indicator to the stack and then setting to clean the second status indicator.

37. (new) The processing device, as recited in claim 35, wherein:

when the second virtual register is mapped to a new physical register, setting the second status indicator to not clean.

38. (new) The processing device, as recited in claim 34, wherein:

a subset of the plurality of virtual registers are virtual local registers.

39. (new) The processing device, as recited in claim 38, further including:

a stack; and

wherein execution of a save command comprises saving the mapping of all of the virtual local registers onto the stack; and saving the status indicators corresponding to all of the virtual local registers onto the stack.

40. (new) The processing device, as recited in claim 39, wherein:

the execution of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators onto the stack.

41. (new) The processing device, as recited in claim 40, wherein:

execution of a restore command comprises popping the mapping of all of the virtual local registers from the stack to the virtual local registers; and popping the status indicators corresponding to all of the virtual local registers from the stack.

42. (new) The processing device, as recited in claim 41, wherein:

if the instruction decoder decodes a return instruction, then executing the restore command.

43. (new) The processing device, as recited in claim 34, wherein:

following the argument binding, if the first virtual register is a destination register, the first virtual register is assigned a first physical register address which is different than a second physical register address stored in the second virtual register.

44. (new) A method comprising:

decoding an instruction;

maintaining a mapping of virtual registers to physical registers, a subset of the virtual registers being virtual local registers;

if the instruction is a save instruction, then executing a save command;

if the instruction is a restore instruction, then executing a restore command; and

wherein

the executing of the save command comprises saving the mapping of all of the virtual local registers onto a stack, and

the executing of the restore command comprises popping the mapping of all of the virtual local registers from the stack to the virtual local registers.

45. (new) The method, as recited in claim 44, wherein:

the executing of the save command further comprises saving status indicators corresponding to all of the virtual local registers onto the stack, and

the executing of the restore command further comprises popping the status indicators corresponding to all of the virtual local registers from the stack.

46. (new) The method, as recited in claim 45, wherein:

the executing of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators corresponding to all of the virtual local registers onto the stack.

47. (new) A processing device including:

an instruction decoder adapted to decode an instruction;

a plurality of physical registers;

a plurality of virtual registers, each of the virtual registers comprising physical register address bits, and a subset of the virtual registers being virtual local registers; and

wherein

if the instruction decoder decodes a save instruction, then executing a save command, the executing of the save command comprising saving a mapping of all of the virtual local registers onto a stack, and

if the instruction decoder decodes a restore instruction, then executing a restore command, the executing of the restore command comprising popping the mapping of all of the virtual local registers from the stack to the virtual local registers.

48. (new) The processing device, as recited in claim 47, further including:

a plurality of status indicators, each of the status indicators corresponding to a respective one of the virtual registers; and

wherein

the executing of the save command further comprises saving the status indicators corresponding to all of the virtual local registers onto the stack, and

the executing of the restore command further comprises popping the status indicators corresponding to all of the virtual local registers from the stack.

49. (new) The processing device, as recited in claim 48, wherein:

the executing of the save command further comprises setting the status indicators corresponding to all of the virtual local registers to clean after the saving of the status indicators corresponding to all of the virtual local registers onto the stack.

50. (new) The processing device, as recited in claim 49,further including:

a plurality of physical register status indicators, each of the physical register status indicators corresponding to a respective one of the physical registers; and

wherein each of the physical register status indicators represents a selected one of a plurality of physical register states, and the physical register states include "free", "waiting", and "valid".

51. (new) The processing device, as recited in claim 50,wherein:

physical registers available for mapping to virtual registers are represented as "free" in the corresponding physical register status indicators.

52. (new) The processing device, as recited in claim 51,wherein:

physical register status indicators transition to representing "waiting" when the corresponding physical registers are mapped to virtual registers.

53. (new) The processing device, as recited in claim 52,wherein:

physical register status indicators transition to representing "valid" when the corresponding physical registers are written.